SRFlow-DA: Super-Resolution Using Normalizing Flow with Deep Convolutional Block

Younghyun Jo Sejong Yang Seon Joo Kim
Yonsei University

Abstract

Multiple high-resolution (HR) images can be generated from a single low-resolution (LR) image, as super-resolution (SR) is an underdetermined problem. Recently, the conditional normalizing flow-based model, SRFlow, shows remarkable performance by learning an exact mapping from HR image manifold to a latent space. The flow-based SR model allows sampling multiple output images from a learned SR space with a given LR image. In this work, we propose SRFlow-DA which has a more suitable architecture for the SR task based on the original SRFlow model. Specifically, our approach enlarges the receptive field by stacking more convolutional layers in the affine couplings, and so our model can get more expressive power. At the same time, we reduce the total number of model parameters for efficiency. Compared to SRFlow, our SRFlow-DA achieves better or comparable PSNR and LPIPS for $\times 4$ and $\times 8$ SR tasks, while having a reduced number of parameters. In addition, our method generates visually clear results without excessive sharpness artifacts.

1. Introduction

The goal of super-resolution (SR) is to generate a high-resolution (HR) image from the given low-resolution (LR) image. Single-image SR is a fundamental task of computer vision and can be used for various applications. A number of deep learning-based SR methods [10, 14, 22, 23] successfully achieve a good SR quality in terms of peak signal-to-noise ratio (PSNR). However, the nice PSNR score does not guarantee the satisfaction of human visual perception [3].

To generate realistic images, generative adversarial network (GAN) [7] is applied for SR the task [13, 21, 18, 25, 24, 28, 19, 17]. The GAN-based SR methods generate plausible high-frequency details and visually satisfactory results. The SR methods have evolved to achieve better quality. However, there have been few considerations to generate a variety of output images, even though the SR task is underdetermined so fundamentally a single LR image can be mapped to many output images. Most of the SR methods generate a single HR output image.

Recently, SRFlow [16] successfully generates diverse SR results by using the normalizing flow [5, 6, 12]. SRFlow extends a flow step used in [12] to receive the LR image as a conditional input for generating an HR image consistent with the LR image. SRFlow maps HR images to a latent space thanks to the tractable loss function of the normalizing flow, and generates diverse HR images from an input random variable sampled from the latent space with the conditional LR image.

In this paper, we introduce a simple but effective modification to the SRFlow architecture for improving the SR quality. The receptive field size of a single flow step in the SRFlow architecture is too small, therefore, we stack more convolutional layers to enlarge the size. Specifically, we stack 6 3×3 convolutional layers in the affine injector and the conditional affine coupling of the SRFlow architecture, and the receptive field size is increased from 5×5 to 13×13 (Fig. 1). In experiments, we found the small receptive field size affects the SR quality both quantitatively and qualitatively. Besides, we reduce the total number of parameters and the model can be trained on a GPU with less memory ($<$11GB). In the NTIRE 2021 Learning the Super-Resolution Space Challenge [15], our approach ranked first place in LR-PSNR which measures how much the SR output is consistent with the LR input when downsampled. To sum up, the contributions of our paper are as follows:

- We tune the SRFlow architecture by simply stacking more convolutional layers to have a large receptive field in a single flow step for better SR quality.
- Compared to the SRFlow, we achieve better or comparable quantitative results in terms of PSNR and LPIPS values while reducing the number of parameters and the training time. Also, our approach generates visually clear output images without excessive sharpness.
- We show experimental results of several variants of our approach. This will motivate researchers to design better flow-based architectures in the future as the normalizing flow is a very new approach for the SR task.

2. Related Work

In this section, we introduce SR methods that generate diverse output images. Among the methods, we use SRFlow as our baseline as it shows good results and can be applied to general images.

2.1. Flow-Based Methods

The normalizing flow was first introduced in [5, 20, 6] to learn an exact mapping from data distribution to a latent distribution by using the tractability of exact log-likelihood. Unlike GAN [7] which learns an implicit density, the normalizing flow explicitly computes the probability density. For the SR task, a conditional normalizing flow method with affine coupling layers was employed in [26]. However, the scale factor was limited to $\times 2$ SR task. Recently, SRFlow [16] applied a similar approach for $\times 4$ and $\times 8$ SR tasks and successfully generated diverse SR results. We will introduce the overview of the SRFlow in Sec. 3.1.

2.2. Other Methods

There are several approaches to learn SR space and output diverse images. VarSR [9] used a variational autoencoder to generate stochastic SR images for faces and numbers. The method trained to match LR and HR latent distributions and an LR image can generate diverse SR images with the reparameterization trick at test time. In Explorable SR [2], a consistency enforcing module was suggested to guarantee the SR output matches the LR input when downsampled, while generating diverse SR images by using the user input. One advantage of this module is that it can be attached to any SR model. DeepSEE [4] used semantic maps and disentangled style codes for explorative SR. The method generated high-quality diverse face images up to $\times 32$ magnification factor.

3. Method

3.1. SRFlow

We use SRFlow [16] as our base model to generate photo-realistic and diverse SR outputs that are consistent with the input LR image. SRFlow is based on the conditional normalizing flow, and we can formally express it as follows:

$$z = f(y; x). \tag{1}$$

An invertible neural network f maps an HR image y to a latent variable z conditioned by the corresponding LR image x. The normalizing flow [5, 20] computes the probability density $p_{y|x}$ explicitly by the log determinant of Jacobian of each invertible layer, and the network is trained by minimizing the negative log-likelihood as follows:

$$L = -\log p(z) - \sum_{n=0}^{N-1} \log |\det \frac{\partial f^n}{\partial h^n}(h^n; g(x))|. \tag{2}$$

where N is the total number of invertible layers, n is the index of each layer, $h^0 = y$, $h^N = z$, $h^{n+1} = f^n(h^n; g(x))$, and g is a deep convolutional neural network that extracts rich feature for the conditioning. For realistic SR, one advantage of using the normalizing flow is that the loss is explicitly represented and practically stable compared to using GAN [7]. After the training, the HR image y can be generated from the latent encoding z as follows:

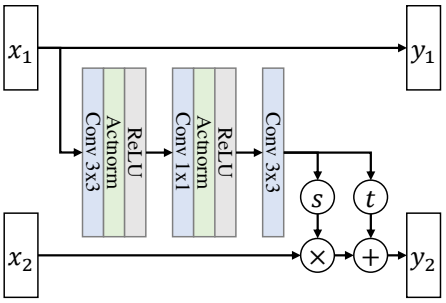
$$y = f^{-1}(z; x), \quad \text{where } z \sim p_z. \tag{3}$$

One matter with the normalizing flow is that all components must be invertible and there are limited components to use including invertible 1×1 convolutional layer [12]. To map HR images well into the latent space (*i.e.* Gaussian distribution), SRFlow stacks a number of flow steps in a multi-scale manner. Each flow step consists of actnorm, 1×1 convolution, affine injector, and conditional affine coupling. Specifically, for $\times 8$ SR, SRFlow stacks 16 flow steps for 4 scale levels (*i.e.* $K = 16$ and $L = 4$). We conjecture that stacking a number of flow steps is required to encourage the expressive power of the model for the SR task, due to the small receptive field size of the components in the flow step. Please refer to the SRFlow paper [16] for the overall details and we will focus on our modifications from the original SRFlow architecture in the next section.

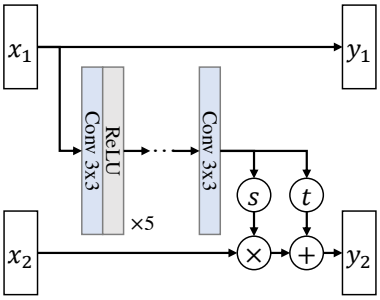
3.2. SRFlow-DA

To improve the SR performance of SRFlow, we focus on enlarging receptive field size in the single flow step. In the affine injector and the conditional affine coupling of SRFlow, a shallow convolutional block is used. The block consists of 3×3 conv, actnorm, ReLU, 1×1 conv, actnorm, ReLU, and 3×3 conv, and its receptive field is 5×5 (Fig. 1a). The main idea of our approach is to stack more convolutional layers to enlarge the receptive field size. Specifically, we stack 6 3×3 convolutional layers followed by ReLU activation except for the last convolutional layer (Fig. 1b), and we remove actnorm by reflecting the convention that using a normalization layer does not much helpful for the SR task [25]. Note that only the actnorms inside the affine layers are removed but the other actnorms as a building block of the flow step still remain. By doing so, the receptive field size is enlarged to 13×13, and the expressive power of the model increases. We can estimate locally accurate affine coefficients s and t, and we empirically found this helps improve the performance. Note that this part is also invertible because the same s and t values can be made by using y_1 at the backward pass and x_2 is computed as $(y_2 - t)/s$. Please refer to [12] for more details.

Compared to the original SRFlow architecture, we reduce the number of flow steps from 16 to 6 ($K = 6$) for each scale level because of the increased receptive field size.



(a) Affine coupling in SRFlow



(b) Our approach (SRFlow-DA)

Figure 1. We stack more convolutional layers in the affine injector and the conditional affine coupling to enlarge the receptive field and enhance the expressive power.

We can naively calculate the receptive field size of the original SRFlow and our approach within a single scale level as 129×129 (16×2 number of conv blocks with 5×5 receptive field size) and 145×145 (6×2 number of conv blocks with 13×13 receptive field size) respectively. In addition, we further reduce the number of scale levels from 4 to 3 ($L = 3$) for ×8 SR and 3 to 2 ($L = 2$) for ×4 SR. Other parts of the architecture are the same as SRFlow, and we use RRDB [25] as LR image feature extractor. In a way, our approach reduces the number of flow steps and the total number of model parameters. As a result, our model can be trained on a single GPU with less than 11GB memory. Note that the original SRFlow model was trained on a single NVIDIA V100 GPU which has at least 16GB memory. We name our approach SRFlow-DA (Deep convolutional block in the Affine couplings).

4. Experiments

4.1. Implementation Details

We follow the same training process as described in the original SRFlow [16], except for the learning rate. We use the train images of DIV2K dataset [1] and Flickr2K dataset. We use HR training patches size of 160×160, and the sizes of the corresponding LR patches are 20×20 and 40×40 for ×8 SR and ×4 SR respectively. We add Gaussian noise with a standard deviation of $\sigma = \frac{4}{\sqrt{3}}$ to the HR patches. We use a pre-trained RRDB network as the LR image encoder g. We first train our SRFlow-DA model f only for 10^5 iterations and then we train both f and g together for further 10^5 iterations. The training is conducted using Adam optimizer [11] with the starting learning rate of 2×10^{-4} and it is halved at 50%, 75%, 90%, and 95% of the total training iterations. After the training, we can generate different SR outputs (Eq. (3)). Specifically, $z \sim \mathcal{N}(0, \tau)$ as we use a Gaussian distribution and we set $\tau = 0.9$ for all our results as did in the original SRFlow model. Please refer to our code for more details.[1]

4.2. Evaluation Metrics

Three metrics are used to quantify the quality of results, which are introduced in the NTIRE 2021 Learning the SR Space Challenge [15]. To measure photo-realism, the learned perceptual image patch similarity (LPIPS) [27] is used, which is designed to measure the quality of images from the perspective of human visual perception. To measure the spanning of the SR space, a diversity score is computed by using 10 generated images from the same input LR image. The global best score is the best one of 10 image-level LPIPS scores, and the local best score is the mean of best pixel-level LPIPS scores. Finally, the diversity score is calculated as (global best - local best) / (global best) * 100. To measure the consistency with the input LR image, the LR-PSNR value is computed with the downsampled output HR image. In addition to the three metrics above, we additionally measure HR-PSNR between the output HR image and the ground truth (GT) image. Note that lower values are better for LPIPS, and higher values are better for the diversity score and PSNR. Please refer to the challenge website for the evaluation code.[2]

4.3. Comparisons with Other Methods

We compare our method on the DIV2K validation set [1] with bicubic interpolation, RRDB [25], ESRGAN [25], and SRFlow [16]. Quantitative results are shown in the upper part of Table 1, and the values are from the SRFlow paper, except for our results (SRFlow-DAs). Compared to SRFlow, for ×4 SR, the number of parameters is reduced from 22.8M to 8.7M and the training time is reduced to 33 hours. The inference time to generate an output image size of 1920×1080 from an input image size of 480×270 is also reduced from 1.98s to 1.18s. Note that the training times of SRFlow-DAs and all the inference times are measured on NVIDIA GeForce RTX 2080 TI. Even though SRFlow-DA reduces the model size and the training time, SR qual-

[1] https://github.com/yhjo09/SRFlow-DA

[2] https : / / github . com / andreas128 / NTIRE21 _ Learning_SR_Space

Method	Params	Time		PSNR		LPIPS			Diversity
		Training	Inference	LR	HR	mean	global	local	
Bicubic	-	-	-	38.70	26.70	0.409	0.409	0.409	0
RRDB	16.7M	-	-	49.20	**29.44**	0.253	0.253	0.253	0
ESRGAN	16.7M	-	-	39.03	26.22	0.124	0.124	0.124	0
SRFlow	22.8M	120h	1.98s	50.64	27.09	**0.120**	**0.119**	**0.089**	25.24
SRFlow-DA	8.7M	33h	1.18s	50.88	27.57	0.121	**0.119**	0.092	23.55
SRFlow-DA-R	8.7M	33h	1.19s	**50.92**	27.23	**0.120**	**0.119**	**0.089**	**25.50**
SRFlow-DA-S	8.8M	25h	0.91s	50.48	27.43	0.130	0.129	0.098	24.01
SRFlow-DA-D	6.4M	30h	1.01s	49.00	26.78	0.132	0.131	0.101	23.24

(a) $\times 4$ SR results on DIV2K validation set.

Method	Params	Time		PSNR		LPIPS			Diversity
		Training	Inference	LR	HR	mean	global	local	
Bicubic	-	-	-	37.16	23.74	0.584	0.584	0.584	0
RRDB	16.7M	-	-	45.43	**25.50**	0.419	0.419	0.419	0
ESRGAN	16.7M	-	-	31.35	22.18	0.277	0.277	0.277	0
SRFlow	34.1M	120h	1.97s	50.09	23.03	0.272	0.276	0.200	**25.28**
SRFlow-DA	13.3M	47h	1.01s	50.91	23.75	0.261	0.259	0.198	23.45
SRFlow-DA-R	13.3M	47h	1.04s	50.73	23.42	**0.256**	**0.253**	**0.191**	24.57
SRFlow-DA-S	15.0M	30h	0.65s	**52.21**	23.71	0.268	0.265	0.204	23.17
SRFlow-DA-D	9.5M	37h	0.80s	45.18	22.54	0.306	0.304	0.241	22.54

(b) $\times 8$ SR results on DIV2K validation set.

Table 1. Quantitative results with other methods on DIV2K validation set. Compared to SRFlow, our approaches (SRFlow-DA and SRFlow-DA-R) achieve better or similar performance with smaller model size and reduced training time. Inference time is measured for generating an output image size of 1920×1080. Note that the training times and the inference times are measured on NVIDIA GeForce RTX 2080 TI. Best values are shown in bold and second best values are underlined.

ity is better in terms of PSNR and comparable in terms of mean LPIPS. Similarly, for $\times 8$ SR, the number of parameters is reduced from 34.1M to 13.3M and the training time is reduced to 47 hours. The inference time to generate an output image size of 1920×1080 from an input image size of 240×135 is reduced from 1.97s to 1.01s. In this case, SRFlow-DA achieves better performance in all metrics except for the diversity score. This is because the diversity score is calculated from the difference of global and local LPIPS values. The difference is larger in SRFlow, however, SRFlow-DA shows better mean, global, and local LPIPS values. We infer the improved performance is from our proposed architecture design which has better expressive power by the non-linearity of deeply stacked conv-relu layers.

Qualitative results for $\times 8$ SR are shown in Fig. 2. RRDB results look overly blurry as it is trained using the mean squared error only. The results show higher PSNR values, but this not guarantee perceptually better images. SRFlow successfully generates photo-realistic images but they often look excessively sharp compared to the GT images in the upper 4 rows. This attribute, however, makes some fine natural textures look realistic such as fur in the lower 2 rows. Compared to SRFlow, our SRFlow-DA results look more clear while maintaining the photo-realism. For the upper 4 rows, SRFlow-DA generates the results that look close to the GT images without excessive sharpness. On the other hand, in the case of the lower 2 rows, SRFlow-DA results look less satisfactory than SRFlow results.

4.4. Challenge Results

In the NTIRE 2021 Learning the SR Space Challenge [15], for $\times 4$ SR task, our results from the SRFlow-DA model obtained 50.70 (1st), 0.121 (3rd), and 23.091 (4th) for LR-PSNR, LPIPS, and the diversity score respectively. Similarly, for $\times 8$ SR task, our results obtained 50.86 (1st), 0.266 (3rd), and 23.320 (4th) respectively. The numbers are calculated on the 100 images of DIV2K test set. Our approach shows the best LR-PSNR with good LPIPS values compared to the other participants. In other words, SRFlow-DA successfully generates visually pleasing diverse output images and maintains the original content of the input LR image.

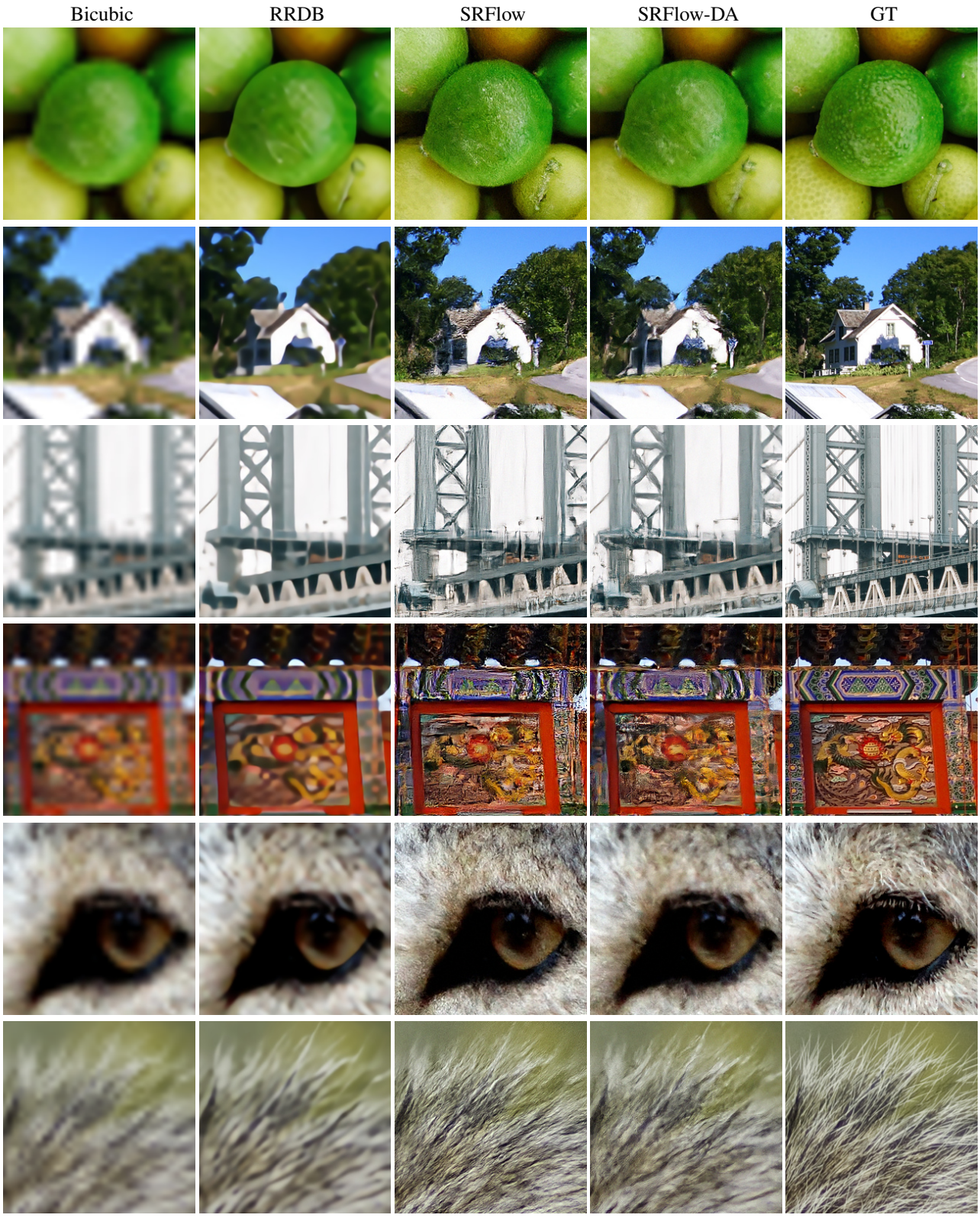
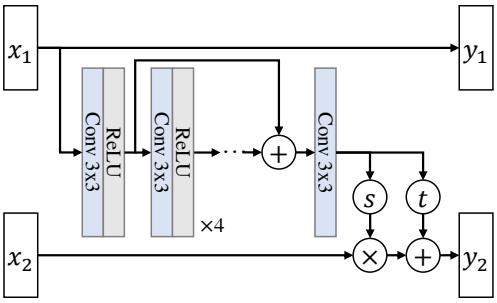
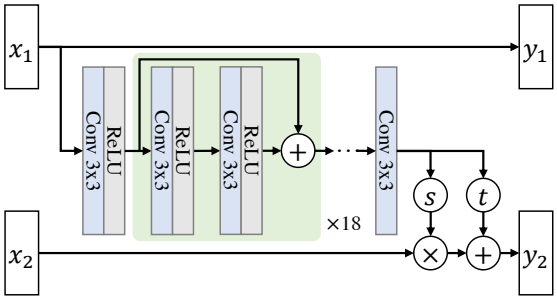


Figure 2. Qualitative results with other methods on the DIV2K validation set for $\times 8$ SR. SRFlow results show a better visual quality but sometimes show excessive sharpness. SRFlow-DA results show clear images while maintaining photo-realism.



(a) SRFlow-DA-R



(b) SRFlow-DA-D

Figure 3. Two variants of SRFlow-DA. See the text for details.

4.5. Variants of SRFlow-DA

4.5.1 SRFlow-DA-R

For SRFlow-DA-R, we add skip connection from the output of the first convolutional layer to the input of the last convolutional layer in our deep convolutional block (Fig. 3a). With the residual learning, the model could learn more rich features.

4.5.2 SRFlow-DA-S

For SRFlow-DA-S, we use a single-scale architecture, like did in conventional deep SR methods. In the case of $\times 8$ SR, we reduce the number of scale levels from 3 to 1 and increase the number of flow steps from 6 to 18 at the same time to balance the number of parameters (*i.e.* $K = 18$ and $L = 1$). We also remove all intermediate squeeze operations, and the remaining first squeeze operation directly resizes the spatial size to $\frac{1}{8}$ which is equivalent to the LR image size. The same thing applies for $\times 4$ SR, and the model is processed at $\frac{1}{4}$ scale in this case.

4.5.3 SRFlow-DA-D

For SRFlow-DA-D, we use more deeper convolutional block in the affine couplings (Fig. 3b). Specifically, we stack 38 3×3 convolutional layers and we add skip connections for every 2 conv-relu layers. Additionally, we reduce the number of flow steps to 1 ($K = 1$). In this setting, we can check if the performance increases using only a deeper block without many flow steps.

4.5.4 Experimental Results

Quantitative results are shown in the lower part of Table 1. In case of $\times 4$ SR, SRFlow-DA-R achieves better LR-PSNR (+0.04), LPIPS (-0.0003), and diversity score (+1.95). In case of $\times 8$ SR, SRFlow-DA-R achieves better LPIPS (-0.0052) and diversity score (+1.12) with lower LS-PSNR (-0.18). Overall, SRFlow-DA-R improves the performance compared to SRFlow-DA. Compared to SRFlow, our SRFlow-DA-R results show better performance in all metrics, reducing the number of parameters and the training time simultaneously. Note again SRFlow-DA-R shows both better global and local LPIPS scores than SRFlow but its diversity score is lower because the score is computed as the span of the two scores. SRFlow-DA-S and SRFlow-DA-D show lower performance than SRFlow-DA, but they have advantages in shorter training and inference time.

The qualitative results of the variants are shown in Fig. 4. SRFlow-DA-R results look clear (row 1) and look closer to the GT than SRFlow-DA (rows 2-4). SRFlow-DA-S results look less sharp than SRFlow-DA-R (rows 4-5). SRFlow-DA-D results look overly noisy than the others. We infer that the reduced number of flow steps makes mapping to the latent space difficult.

In addition, diverse output generation results are shown in Fig. 5. We plot 4 output patches of the same input image for SRFlow-DA and SRFlow-DA-R. Different high-frequency details are generated in each output image. SRFlow-DA results look slightly noisy than SRFlow-DA-R (rows 1-2). We infer the skip connection in SRFlow-DA-R helps the training [8]. Still, both methods have difficulty in restoring thin lines (rows 1, 3).

5. Conclusion

We proposed a simple modification of SRFlow for better SR quality while reducing the model size and the training time (SRFlow-DA). The receptive field of a single flow step is increased by using a deep convolutional block inside the affine coupling, and we empirically found this is a useful approach for the SR task. Through the experiments, we verified SRFlow-DA achieves better PSNR and LPIPS values by preserving the original contents, but the diversity score little decreased. In addition, SRFlow-DA achieved good scores in the NTIRE 2021 Learning the SR Space Challenge [15], for both $\times 4$ and $\times 8$ SR tracks. We believe that our approach can motivate researchers to design a better normalizing flow-based SR architecture in the future.

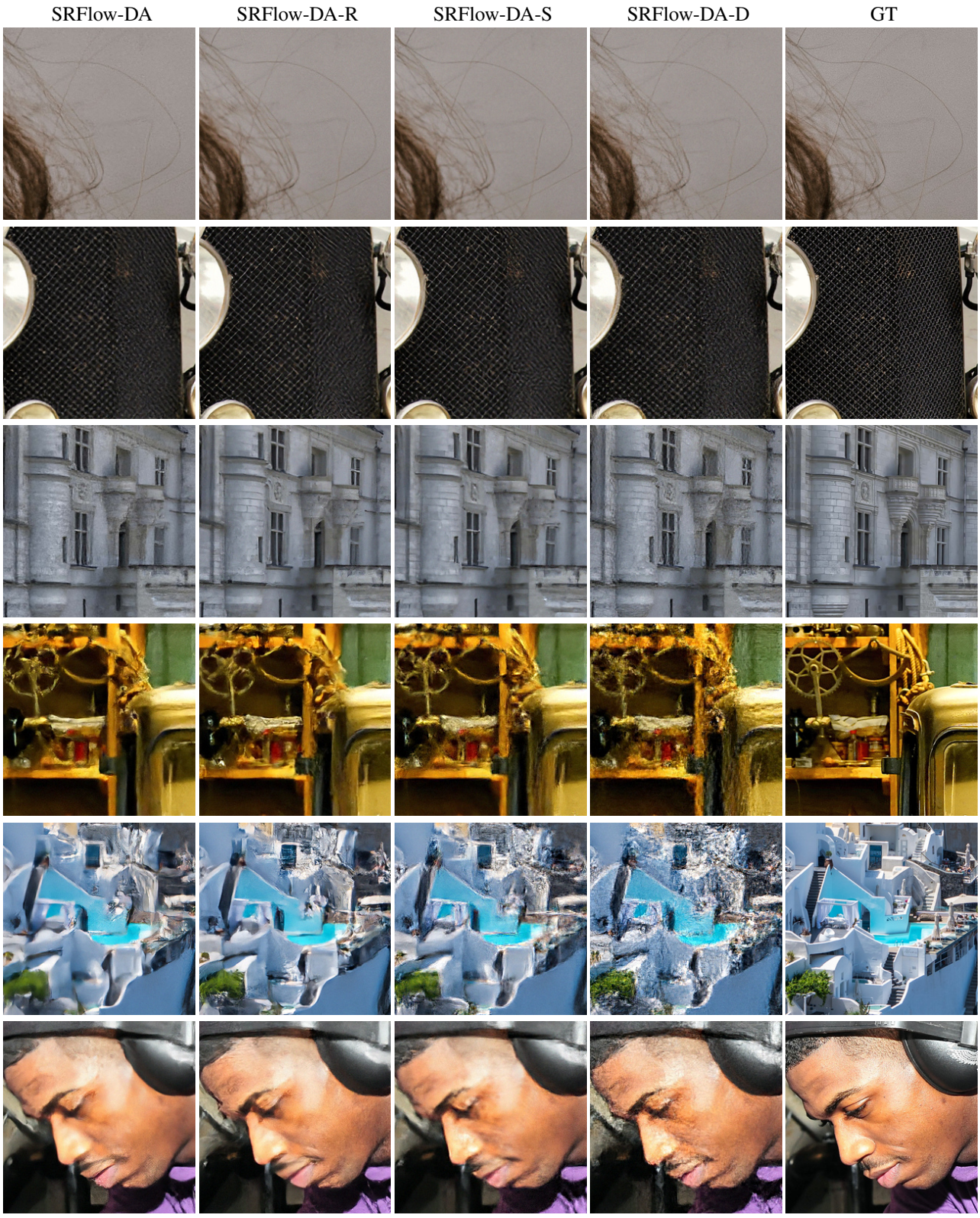


Figure 4. Visual comparisons for the variants of SRFlow-DAs on the DIV2K validation set. The first 3 rows are $\times 4$ SR results and the last 3 rows are $\times 8$ SR results. Overall, SRFlow-DA-R results show clear images with well-generated textures.

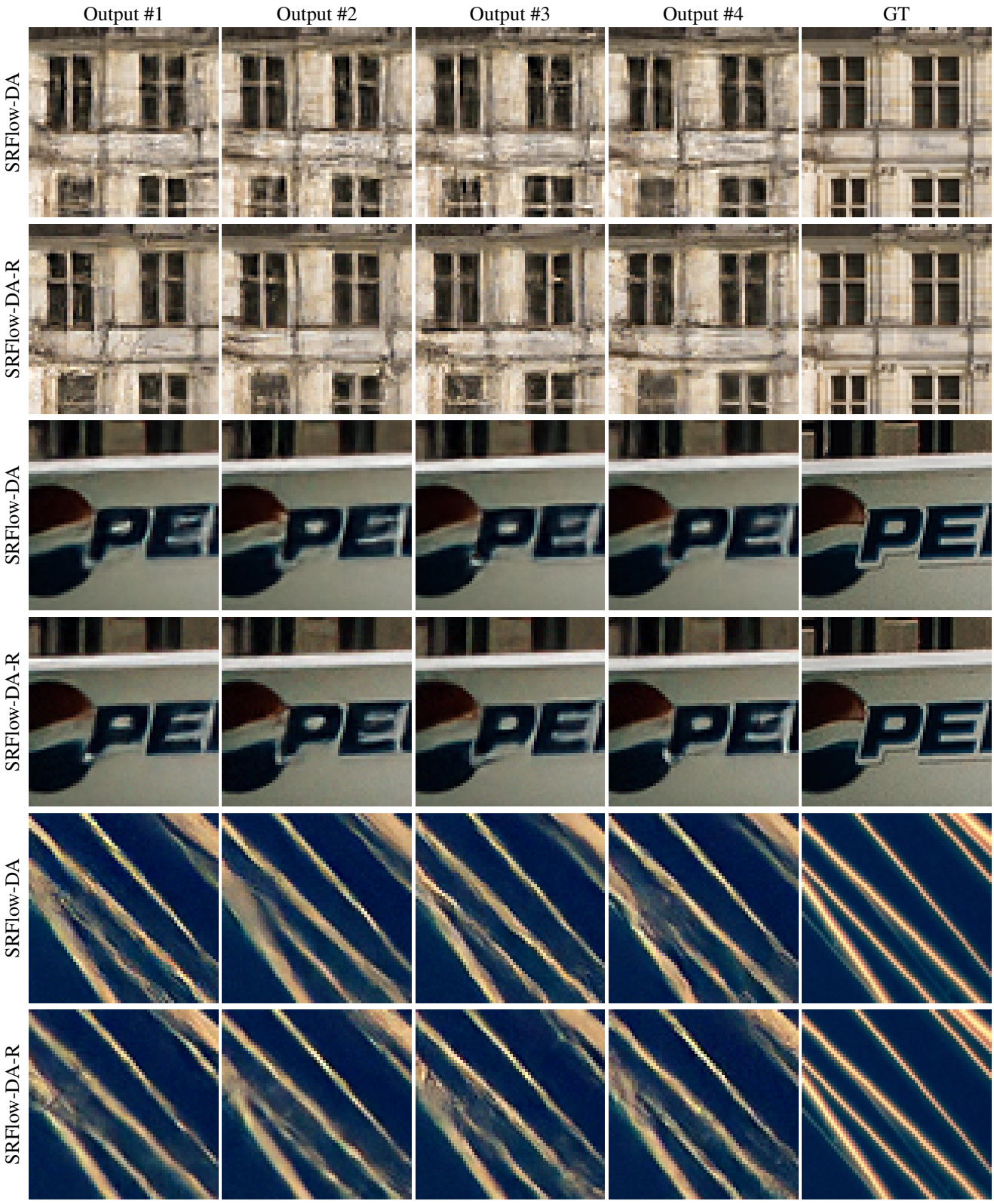


Figure 5. Diverse SR results of SRFlow-DA and SRFlow-DA-R on the DIV2K validation set. The methods successfully generate diverse output images, and SRFlow-DA-R results look slightly more consistent with the GT images.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshops*, July 2017. 3

[2] Yuval Bahat and Tomer Michaeli. Explorable super resolution. In *CVPR*, pages 2713–2722. IEEE, 2020. 2

[3] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6228–6237, 2018. 1

[4] Marcel C. Bühler, Andrés Romero, and Radu Timofte. Deepsee: Deep disentangled semantic explorative extreme super-resolution. In *ACCV*, volume 12625 of *Lecture Notes in Computer Science*, pages 624–642. Springer, 2020. 2

[5] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 1, 2

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 1, 2

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 1, 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[9] Sangeek Hyun and Jae-Pil Heo. Varsr: Variational super-resolution network for very low resolution images. In *ECCV*, 2020. 2

[10] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. 1

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[12] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NeurIPS*, volume 31. Curran Associates, Inc., 2018. 1, 2

[13] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, July 2017. 1

[14] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, pages 136–144, 2017. 1

[15] Andreas Lugmayr, Martin Danelljan, Radu Timofte, et al. Ntire 2021 learning the super-resolution space challenge: Methods and results. *CVPR Workshops*, 2021. 1, 3, 4, 6

[16] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, pages 715–732. Springer, 2020. 1, 2, 3

[17] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *CVPR*, 2020. 1

[18] Seong-Jin Park, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong, and Seungyong Lee. Srfeat: Single image super-resolution with feature discrimination. In *ECCV*, pages 439–455, 2018. 1

[19] Mohammad Saeed Rad, Behzad Bozorgtabar, Urs-Viktor Marti, Max Basler, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Srobb: Targeted perceptual loss for single image super-resolution. In *ICCV*, October 2019. 1

[20] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, pages 1530–1538. PMLR, 2015. 2

[21] Mehdi S. M. Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, Oct 2017. 1

[22] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *CVPR*, pages 3118–3126, 2018. 1

[23] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *CVPR*, pages 3516–3525, 2020. 1

[24] Jae Woong Soh, Gu Yong Park, Junho Jo, and Nam Ik Cho. Natural and realistic single image super-resolution with explicit natural manifold discrimination. In *CVPR*, June 2019. 1

[25] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV Workshops*, September 2018. 1, 2, 3

[26] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019. 2

[27] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 3

[28] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *ICCV*, October 2019. 1